JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 05346858
(43)Date of publication of application: 27.12.1993

(51)Int.Cl. G06F 9/45

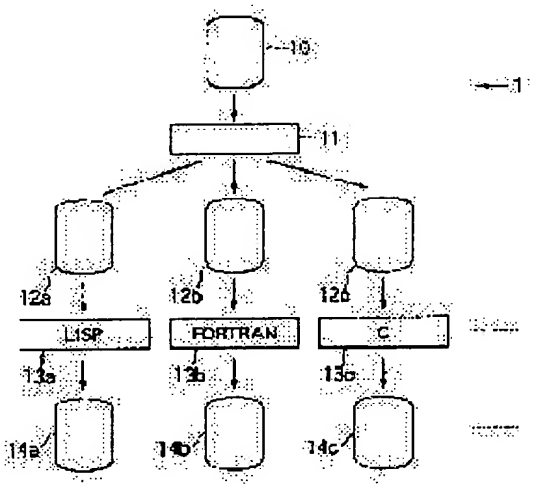
(21)Application number: 04179304
(22)Date of filing: 12.06.1992

(71)Applicant: SONY CORP
(72)Inventor: MAEKAWA HIROTOSHI
YASUDA HIROYUKI

(54) PROGRAM EDITING METHOD

(57)Abstract:

PURPOSE: To provide the program editing method which can efficiently develop a program with a single program language concerning the program conventionally developed with each of plural program languages corresponding to functions for each section in a digital computer system.
CONSTITUTION: A first compiler 11 analyzes a source program 10 containing plural functions provided with the functions of difference use for each function, and the source program is divided and converted into next-stage programs 12 (12a-12c) described in the plural program languages corresponding to the use. Further, the next-stage programs 12 are converted to machine wood programs 14 (14a-14c) by second compilers 13 (13a-13c) corresponding to the respective program languages. Moreover, such a method is applied to a cross compiler environment as well, and a program used for plural computers is obtained by editing a single program on a single computer.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

MENU SEARCH INDEX

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開平5－346858
(43)公開日　平成５年(1993)12月27日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　9/45					
		9292－5B	Ｇ０６Ｆ　9/44	３２２　Ａ	
		9292－5B		３２０　Ａ	

審査請求　未請求　請求項の数４(全　８　頁)

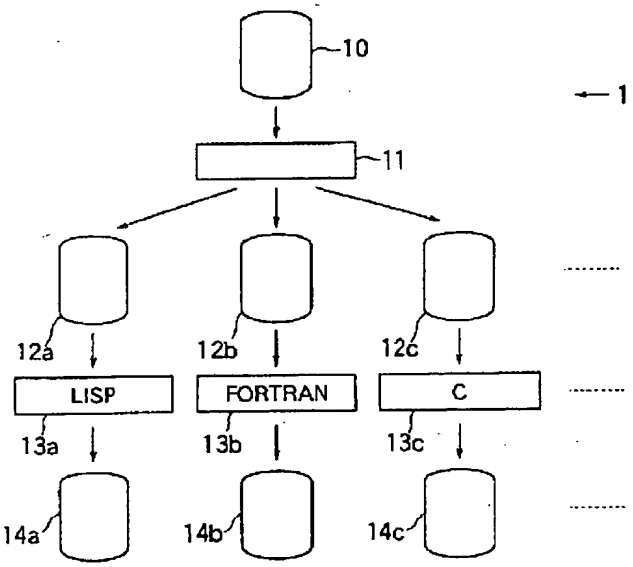
(21)出願番号　特願平4－179304

(22)出願日　平成４年(1992)６月12日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　前川　博俊
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　安田　弘幸
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　弁理士　佐藤　隆久

(54)【発明の名称】　プログラム編集方法

(57)【要約】
【目的】　デジタル計算機システムにおいて、従来機能に応じて部分ごとに複数のプログラム言語で開発されていたプログラムを単一のプログラム言語で能率のよいプログラムの開発の可能なプログラム編集方法を得る。
【構成】　用途の異なった機能を持つ複数の関数を含むソースプログラム１０を関数ごとに、第一のコンパイラ１１で解析し、用途に合った複数のプログラム言語で記述された次段階プログラム１２に分割、変換する。更に、次段階プログラム１２をそれぞれのプログラム言語に対応した第二のコンパイラ１３により機械語プログラム１４に変換される。さらに、上記のような方法をクロスコンパイラ環境についても応用し、複数の計算機に使用するプログラムを、単一の計算機上で単一のプログラムを編集することにより得る。



【特許請求の範囲】
【請求項１】一種類の高級プログラミング言語で記述されたソースプログラムについて、
複数の高級プログラム言語の編集機能を有し、
上記ソースプログラムの部分ごとの処理内容を分析し、
その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、
上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、
さらに、上記変換されたプログラムを機械語に変換することを特徴としたプログラム編集方法。
【請求項２】請求項１のプログラム編集方法において、
上記処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることを特徴とするプログラム編集方法。
【請求項３】請求項２のプログラム編集方法において、
複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、
上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することを特徴とするプログラム編集方法。
【請求項４】請求項３のプログラム編集方法において、
一の計算機おいて、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、
上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、
他の計算機において、さらに、上記変換されたプログラムを機械語に変換することを特徴としたプログラム編集方法。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、デジタル計算機におけるコンパイラに関するものであり、特に複数のプログラム言語が同一の計算機システムで動作する場合におけるコンパイラおよびプログラム編集方法に関する。
【０００２】
【従来の技術】従来からＦＯＲＴＲＡＮ等の高級プログラム言語で記述されたソースプログラムを機械語等の下位言語に変換するコンパイラといったプログラム編集方法が知られ、使用されている。以下、コンパイラを使用し、高級プログラム言語で記述されたソースプログラムを下位言語に変換する、従来のプログラム編集方法について述べる。図９は従来のコンパイラでのデータの流れを示した図である。ソースプログラム１２ｂはＦＯＲＴＲＡＮで記述されたプログラムである。コンパイラ１３ｂはＦＯＲＴＲＡＮで記述されたソースプログラムの文法チェック等を行い、機械語に翻訳する処理（コンパイル）を行うプログラムである。オブジェクトプログラム１４ｂは、ソースプログラム１２ｂをコンパイラ１３ｂにより機械語に翻訳したプログラムである。以上の各プログラムはいずれも同一の計算機上の記憶装置上にファイルの形で存在するものである。
【０００３】計算機使用者は、エディタと呼ばれる文書作成用プログラムを使用して先ずソースプログラム１２ｂをＦＯＲＴＲＡＮで作成する。計算機使用者は、ソースプログラム１２ｂの作成が終了した後、コンパイルを行うソースプログラム１２ｂを指定し、コンパイラ１３ｂを起動する。起動されたコンパイラ１３ｂはソースプログラム１２ｂの文法チェック、変数チェック等を行った後、ソースプログラム１２ｂを機械語に翻訳しオブジェクトプログラム１４ｂを作成する。
【発明が解決しようとする課題】
【０００４】従来のプログラム編集方法は以上のように構成されているため、一種類のコンパイラは一種類の高級プログラム言語しか取り扱うことができなかった。高級プログラム言語を使用した場合、ほとんどの用途のプログラムについて単一言語で対応することができる。一方、各高級プログラム言語はそれぞれ用途に向き、不向きがあり、例えば、ＦＯＲＴＲＡＮは数値演算に向いているが、計算機のハードウェアに密着した入出力（Ｉ／Ｏ）を取り扱うといった用途には向いていない。このため、例えば、ＦＯＲＴＲＡＮを使用して数値演算とＩ／Ｏ操作の混在したプログラムを作成すると、数値計算処理は高速だが、Ｉ／Ｏ操作処理は遅いオブジェクトプログラムが生成されるという問題が生じる。
【０００５】上記問題を解決するためには、計算機使用者がソースプログラムの仕様を分析し、数値演算部分をＦＯＲＴＲＡＮで記述し、Ｉ／Ｏ操作部分を、例えばＣで記述する方法がある。この方法を用いた場合、ソースプログラム仕様を分析し、各部分について異なった高級プログラム言語で記述するという高度な能力が計算機使用者に要求され、ソースプログラム開発上の大きなネックが生じるという問題点があった。本発明は、このような従来技術の問題点に鑑みてなされたものであり、単一のソースプログラムから、自動的にその部分ごとの処理に最適な高級プログラム言語を選択し、その高級プログラム言語で記述された次段階のプログラムに変換することができ、また、上記変換が複数の計算機からなる計算機ネットワークにも適用でき、しかも上記変換処理を行う際に、計算機ネットワーク全体の資源を有効利用することができるプログラム編集方法を提供することを目的とする。
【０００６】
【課題を解決するための手段】上記の問題を解決するため、本発明に係るプログラム編集方法は、一種類の高級プログラミング言語で記述されたソースプログラムについて、複数の高級プログラム言語の編集機能を有し、上

記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換することを特徴とする。また、処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることを特徴とする。

【０００７】また、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することを特徴とする。また、一の計算機おいて、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、他の計算機において、さらに、上記変換されたプログラムを機械語に変換することを特徴とする。

【０００８】

【作用】一種類の高級プログラミング言語で記述されたソースプログラムについて、複数の高級プログラム言語の編集機能を有し、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換することにより、処理内容に応じて最適な高級プログラム言語を選択することを可能とし、また、変換後のプログラムの計算機間のポータビティを実現することを可能としている。また、処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることにより、標準的な各高級プログラム言語から機械語への変換用コンパイラより高度な機能および性能を持つ、上記計算機専用に作成されたコンパイラの使用を可能としている。

【０００９】また、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することにより、上記計算機ネットワーク全体として最適なオブジェクトプログラムを得ることを可能としている。また、一の計算機おいて、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、他の計算機において、さらに、上記変換されたプログラムを機械語に変換することにより、上記計算機ネットワーク全体としての計算機資源の有効利用が可能となる。

【００１０】

【実施例】図面を参照して本発明の第一の実施例について説明する。図２は、図１（Ａ）に示されたプロセッサエレメント（ＰＥ）の構成を示す図である。本発明は、図２に示すような記号処理用計算機（図２におけるＥＵ）とマンマシンインターフェースおよび記憶等に使用される計算機（図２におけるＲＭ）からなる並列計算機ネットワークのＰＥに適用される。エバリュエーター（ＥＵ）とリソースマネージャー（ＲＭ）はＲＭインタフェース（ＲＭ　Ｉ／Ｆ）で接続され、ＥＵからＲＭに機能データを送出する。また、ＥＵからメインメモリインターフェース（ＭＭ　Ｉ／Ｆ）を介して情報データが送出され、メインメモリ（ＭＭ）とＲＭに送出される。ＲＭ　Ｉ／ＦおよびＭＭ　Ｉ／ＦはＥＵ内の内部バス（ＩＢＵＳ）を介してＡＬＵに接続される。また、ＲＭ　Ｉ／ＦはＲＭ内のエバリュエータインターフェース（ＥＵ　Ｉ／Ｆ）を介してＷＳに接続される。ＥＵはＡＬＵ、マスカ、シフタ、ローカルメモリ、レジスタファイル、ユーザスタック、ディスパッチテーブル、インストラクションキャッシュ、システムコントローラ、コントロールメモリ、ＩＢＵＳ、ダイアグノスティクススタティクス、ＲＭ　Ｉ／ＦおよびＭＭ　Ｉ／Ｆから構成され、ＲＭはＷＳＲ３０００、ダイアグノスティクスインターフェース、通信インターフェース、二次記憶、ＭＭ、ポインタマニュピレータ、ＥＵ　Ｉ／Ｆ、ＣＲＴディスプレイから構成されている。

【００１１】図３は本実施例を説明するための図２の計算機ネットワークの本発明に関する部分を要約したものである。図３において、第一の計算機２は、図２のＲＭに、ＣＰＵ２０はＷＳに、メモリ２２は、図２のＭＭに相当に、表示装置２４は図２のＣＲＴディスプレイに、通信インターフェース２５は図２のＲＭ　Ｉ／Ｆに相当する。磁気記憶装置２６は図２の二次記憶に相当する。なお、図２においては、バス２１およびＩ／Ｏ２３は省略されている。

【００１２】図３において、ＣＰＵ２０は、メモリ２２に蓄積されたプログラムの実行を行い、各種演算処理、および、第一の計算機２の制御を行うＣＰＵおよびその周辺回路である。バス２１は、第一の計算機２の各部分間の情報を伝達するバスである。メモリ２２は、プログラムおよびデータの記憶を行うメモリである。Ｉ／Ｏ２３は、表示装置２４、通信インターフェース２５および磁気記憶装置２６の操作を行うＩ／Ｏ制御装置である。表示装置２４は、第一の計算機２の出力データを表示するＣＲＴディスプレイである。通信インターフェース２５は他の計算機との通信を行うインターフェースである。磁気記憶装置２６は、データの記憶を行う記憶装置である。

【００１３】図４は本発明のプログラム編集方法のプログラム変換方法を示す図である。ソースプログラム１０は、ＬＩＳＰで記述されたソースプログラムである。第

一のコンパイラ１１は、ソースプログラム１０をＬＩＳＰ、ＦＯＲＴＲＡＮ、Ｃ等の各高級プログラム言語に変換するコンパイラである。次段階プログラム１２ａ、ｂ、ｃは、ソースプログラム１０の各部分部分について、第一のコンパイラ１１が生成した各高級プログラム言語、ＬＩＳＰ、ＦＯＲＴＲＡＮおよびＣによるプログラムである。第二のコンパイラ１３ａ、ｂ、ｃは、次段階プログラム１２ａ、ｂ、ｃを機械語に変換するコンパイラである。機械語プログラム１４ａ、ｂ、ｃは、第一のコンパイラ１１および第二のコンパイラ１３ａ、ｂ、ｃがソースプログラム１０を変換することにより生成した機械語プログラムである。以上のソースプログラム１０および各コンパイラは第一の計算機２の磁気記憶装置２６上に記憶され、必要に応じてメモリ２２上に読みだされる。次段階プログラム１２ａ、ｂ、ｃおよび機械語プログラム１４ａ、ｂ、ｃは、第一のコンパイラ１１および第二のコンパイラ１３ａ、ｂ、ｃの動作に従い、順次生成され、メモリ２２または磁気記憶装置２６に記憶される。

【００１４】以下、本発明のプログラム編集方法の動作について説明する。図５は本発明のプログラム編集方法のプログラム変換方法の処理のフローチャートである。ステップ０１（Ｓ０１）において、第一のコンパイラ１１はソースプログラム１０の文法チェック等を行った後、ソースプログラム１０に含まれる各関数ごとに解析を行う。ソースプログラム１０は図６に示すような機能を実現する関数１０ａ～ｄを含んでいるものとする。ここで、関数１０ａは、データＡを通信インターフェース２５に接続される他の計算機から受信するための関数、関数１０ｂは、データＡについて、記号処理を行うための関数、関数１０ｃは、データＡについて、数値演算処理を行うための関数、関数１０ｄは、以上の処理結果を表示装置２４に表示し、磁気記憶装置２６に記憶するための関数であるとする。第一のコンパイラ１１はソースプログラム１０の各関数１０ａ～ｄに含まれる命令の種類を分類し、各関数のリスト処理用命令数、数値計算用の命令数およびＩ／Ｏ用の命令数をカウントする。ここで、関数１０ａはリスト処理用命令を３、数値計算用の命令を１０、Ｉ／Ｏ用の命令を５０含み、関数１０ｂはリスト処理用命令を２００、数値計算用の命令を４０、Ｉ／Ｏ用の命令を０含み、関数１０ｃはリスト処理用命令を２０、数値計算用の命令を４００、Ｉ／Ｏ用の命令を０含み、関数１０ｄはリスト処理用命令を１０、数値計算用の命令を１０、Ｉ／Ｏ用の命令を５００含んでいるという結果が生じたものとする。

【００１５】ステップ０２（Ｓ０２）において、第一のコンパイラ１１はＳ０１の処理の結果求められた命令数の結果に基づき順次、関数１０ａを、Ｉ／Ｏ操作に適したプログラム言語であるＣに変換するためステップ０３に進む。関数１０ｂを、リスト処理に適したプログラム言語であるＬＩＳＰに変換するためステップ０４に進む。関数１０ｃを、数値演算処理に適したプログラム言語であるＦＯＲＴＲＡＮに変換するためステップ０４に進む。関数１０ｄを、Ｉ／Ｏ操作に適したプログラム言語であるＣに変換するためステップ０４に進む。という処理を行う。

【００１６】ステップ０３（Ｓ０３）においては、第一のコンパイラ１１がソースプログラム１０より関数１０ｂの部分を切り出し、次段階プログラム１２ａを得る。ここでは、ソースプログラム１０がＬＩＳＰで記述されているため、関数を切り出し、他の関数とのインターフェースに関する処理を行うのみで、変換は行わない。ステップ０４（Ｓ０４）においては、第一のコンパイラ１１がソースプログラム１０より関数１０ｃの部分を切り出し、ＦＯＲＴＲＡＮへの変換を行い、第二の次段階プログラム１２ｂを得る。ステップ０５（Ｓ０５）においては、第一のコンパイラ１１がソースプログラム１０の関数１０ａおよび関数１０ｄの部分を切り出し、Ｃへの変換を行い、次段階プログラム１２ｃを得る。ステップ０５（Ｓ０５）において、全関数について処理を終わったかを判断する。

【００１７】以上の動作によって得られた次段階プログラム１２ａ～ｃは、それぞれＬＩＳＰ用コンパイラ１３ａ、ＦＯＲＴＲＡＮ用コンパイラ１３ｂ、Ｃ用コンパイラ１３ｃにより、機械語プログラム１４ａ～ｃに変換される。以上により、ソースプログラム１０の各関数１０ａ～ｄについて最適化された、第一の計算機２用の次段階プログラム１２ａ、ｂ、ｃおよび機械語プログラムａ、ｂ、ｃが得られる。ここで、本実施例で使用される高級プログラム言語、ＬＩＳＰ、ＦＯＲＴＲＡＮおよびＣの仕様は、情報処理の分野の標準的な規格に完全準拠したものである。必要な場合、他の計算機に本実施例のプログラム編集方法で生成された次段階プログラムを容易に移植可能である。

【００１８】以下、第二の実施例について述べる。図７は本実施例が適用される、図２の計算機ネットワークの要点を抜き出した図である。第二の計算機３は、図２のＥＵに相当する。第二の計算機３は記号処理を高速に行うための専用計算機として使用される。第二の計算機３の構成は説明の簡略化のために、第一の計算機２と同様のものであると仮定する。ＲＭ　Ｉ／Ｆ３０およびＭＭ　Ｉ／Ｆ３１は第一の計算機２と第二の計算機３を接続し、各種情報を送受信するためのインターフェースである。第一の計算機２と第二の計算機３は、このいずれかを介して必要な情報の送受信を行う。

【００１９】第一の計算機２および第二の計算機３は、それぞれ第一の実施例に示した本発明に係るプログラム編集方法を実施するための計算機環境を持っているものとする。ただし、第二の計算機３の第一のコンパイラ１１、ＬＩＳＰ用コンパイラ１３ａおよびＦＯＲＴＲＡＮ

用コンパイラ１３ｂは、第二の計算機３用に特化し、強化された独特の命令に対応したものであるとする。上記両計算機において、それぞれ第一の計算機２用に開発されたソースプログラム１０および第二の計算機３用に開発されたソースプログラム１０を第一の実施例と同様の方法で機械語プログラム１４ａ、ｂ、ｃに変換する。以上により、計算機の種類に応じて、最も最適化された機械語プログラム１４ａ、ｂ、ｃを得ることができる。

【００２０】以下、第三の実施例について述べる。本実施例はいわゆるクロスコンパイル環境を提供するものである。図８は本実施例のプログラム編集方法のプログラム変換方法を示す図である。本実施例のプログラム編集方法の動作計算機環境は、図７の第一の計算機２に設けられる。ソースプログラム１０は、ＬＩＳＰで記述され、第一の計算機２および第二の計算機３上で動作する関数が混在し、その関数ごとに、一定のフォーマットで前記いずれの計算機で動作するかを設定してあるものとする。第一のコンパイラ１１ａは、ソースプログラム１０の上記設定に基づき、ソースプログラム１０の各部分を第一の計算機２用と第二の計算機３用に分割する。ソースプログラム１０ａ、ｂは、それぞれ第一の計算機２用、第二の計算機３用に分割されたソースプログラム１０である。第一のコンパイラ１１ａ、ｂは、それぞれ第一の計算機２用、第二の計算機３用に特化した第一のコンパイラ１１である。第一のコンパイラ１１ｂは第二の計算機３用に強化した各プログラム言語の命令に対応しているのとする。次段階プログラム１２ａ、ｂ、ｃは第一の計算機２用の次段階プログラムである。次段階プログラム１２ｄ、ｅ、ｆは第二の計算機３用の次段階プログラムである。第二のコンパイラ１３ａ、ｂ、ｃは、第一の計算機２用のＬＩＳＰ用、ＦＯＲＴＲＡＮ用、Ｃ用コンパイラである。第二のコンパイラ１３ｄ、ｅ、ｆは、第二の計算機３用のＬＩＳＰ用、ＦＯＲＴＲＡＮ用、Ｃ用コンパイラである。これらは、第二の計算機３用に強化された命令に対応しているものとする。

【００２１】次に動作を説明する。ソースプログラム１０は、第一のコンパイラ１１ｂにより、第一の計算機２用および第二の計算機３用の各部分に分割される。この際、第一のコンパイラ１１ｂはソースプログラム１０作成時に挿入された上記区別に従い、関数ごとに分割するものとする。この結果、ソースプログラム１０ａ、ｂが生成される。このソースプログラム１０ａ、ｂについて、第一の実施例と同様の手順により最終的な機械語プログラム１４ａ～ｆが生成される。このうち機械語プログラム１４ｄ～ｆは第二の計算機３用のものである。機械語プログラム１４ｄ～ｆは第二の計算機３上に移され、動作する。本実施例では、予め上記区別をソースプログラム１０に挿入したが、第一のコンパイラ１１ａにおいて、自動的に区別するように構成してもよい。また、最初からソースプログラム１０ａ、ｂを別々に開発してもよい。

【００２２】以下、第四の実施例を説明する。図８において、第一の計算機２には第二のコンパイラ１３ｄ～ｆを除いたプログラム編集方法計算機環境を、第二の計算機３には第二のコンパイラ１３ｄ～ｆのみを用意する。第一の計算機２においては、第一の計算機２用の機械語プログラム１４ａ、ｂ、ｃを第一の実施例と同様に作成する。また、第三の実施例と同様の方法で第二の計算機３用の次段階プログラム１２ｄ～ｆを作成し、第二の計算機３に移す。第二の計算機３上では、次段階プログラム１２ｄ～ｆを第二のコンパイラ１３ｄ～ｆにかけ、最終的な機械語プログラム１４ｄ～ｆを作成する。

【００２３】本発明のプログラム編集方法は、上記実施例に限定位されず、他に種々の構成をとることができる。また、上記した計算機環境は例示である。特に、計算機の数、種類、第一のコンパイラ１１の生成する次段階プログラム１２に使用されるプログラム言語、ソースプログラム１０のプログラム言語はここで述べたものに限らない。

【００２４】

【発明の効果】以上述べたように本発明のプログラム編集方法によれば、一種類の高級プログラミング言語で記述されたソースプログラムについて、複数の高級プログラム言語の編集機能を有し、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換することにより、処理内容に応じて最適な高級プログラム言語を選択することを可能とし、また、変換後のプログラムの計算機間のポータビティを実現したプログラム編集方法を提供する。また、処理内容に応じた変換処理が、適用される計算機の種類に応じて行われることにより、標準的な各高級プログラム言語から機械語への変換用コンパイラより高度な機能および性能を持つ、上記計算機専用に作成されたコンパイラの使うことができるプログラム編集方法を提供する。

【００２５】また、複数の計算機から構成され、上記処理内容ごとに異なった計算機を使用する計算機ネットワークにおいて、上記各計算機に対応した、上記次段階のプログラム言語で記述されたプログラムを作成することにより、上記計算機ネットワーク全体として最適なオブジェクトプログラムを得ることができるプログラム編集方法を提供する。また、一の計算機おいて、上記ソースプログラムの部分ごとの処理内容を分析し、その処理内容に応じて、上記複数のプログラム言語の内、その部分について最適な次段階のプログラム言語を選択し、上記部分ごとに、上記ソースプログラムを選択されたプログラム言語に変換し、他の計算機において、さらに、上記変換されたプログラムを機械語に変換することにより、

上記計算機ネットワーク全体としての計算機環境の有効利用を図ることができるプログラム編集方法を提供する。

【図面の簡単な説明】

【図１】本発明が適用される計算機ネットワークの接続形態を示す図である。

【図２】本発明が適用される計算機ネットワークの構成図である。

【図３】本発明が適用される計算機の構成図である。

【図４】本発明のプログラム編集方法のプログラム変換方法を示す図である。

【図５】本発明のプログラム編集方法のプログラム変換方法の処理のフローチャートである。

【図６】ソースプログラムの構成を示す図である。

【図７】本発明の第三実施例の計算機ネットワークの構成を示す図である。

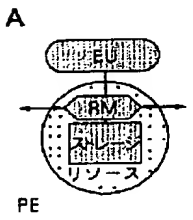
【図８】本発明の第三の実施例のプログラム編集方法のプログラム変換方法を示す図である。

【図９】従来のプログラム編集方法を示す図である。

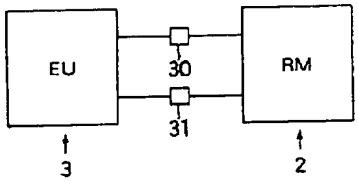
【符号の説明】

１・・・本発明のプログラム編集方法
２・・・本発明が適用される第一の計算機
３・・・本発明が適用される第二の計算機
１０・・・ソースプログラム
１１・・・第一のコンパイラ
１２・・・次段階の高級プログラム言語によるプログラム
１３・・・第二のコンパイラ
１４・・・最終的な機械語プログラム
２０・・・ＣＰＵ
２１・・・バス
２２・・・メモリ
２３・・・Ｉ／Ｏ
２４・・・表示装置
２５・・・通信インターフェース
２６・・・磁気記憶装置
３０・・・ＲＭ　Ｉ／Ｆ
３１・・・ＭＭ　Ｉ／Ｆ

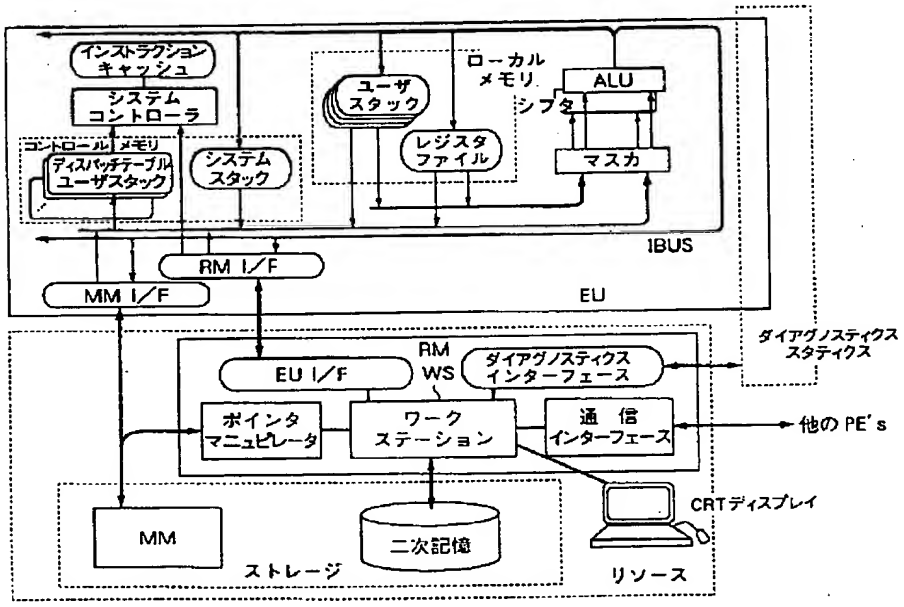
【図１】



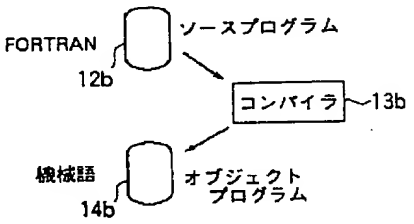
【図７】



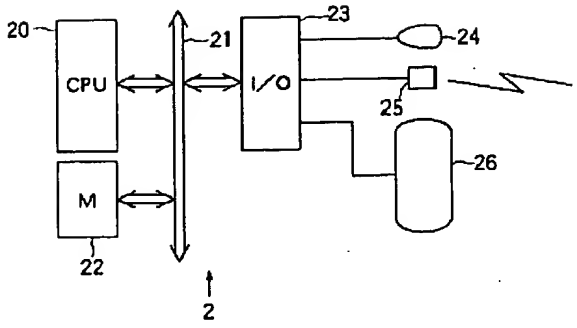
【図２】



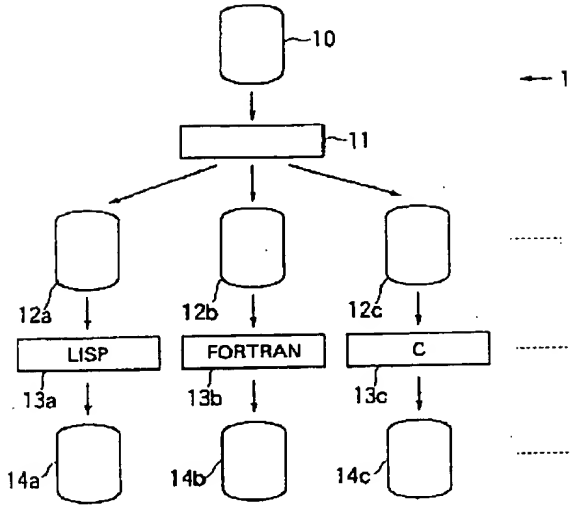
【図９】



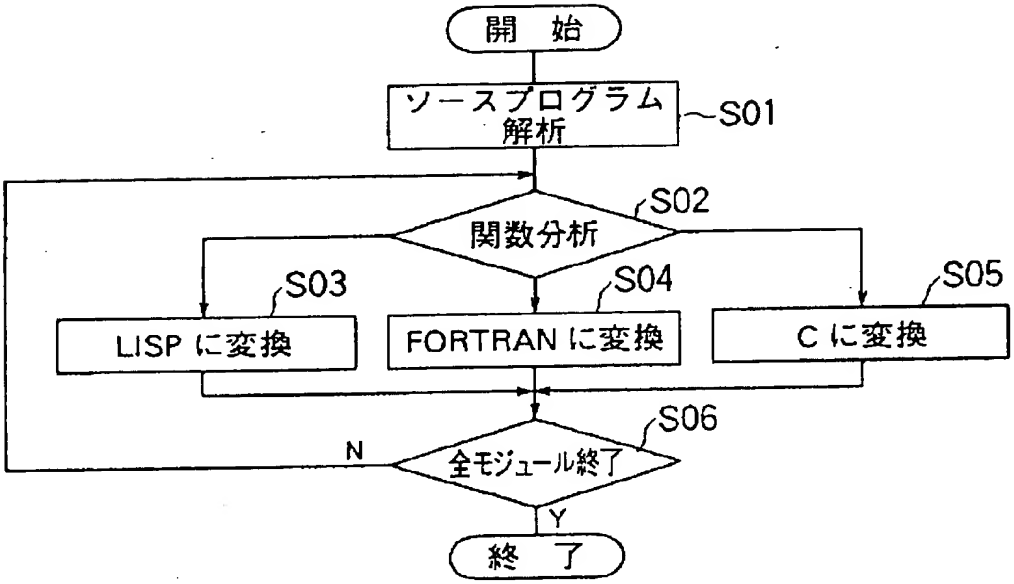
【図３】


20
CPU
21
I/O
23
24
25
26
M
22
2

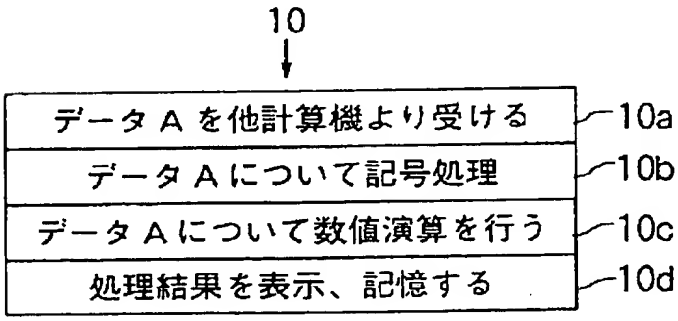
【図４】


10
1
11
12a
12b
12c
LISP
FORTRAN
C
13a
13b
13c
14a
14b
14c

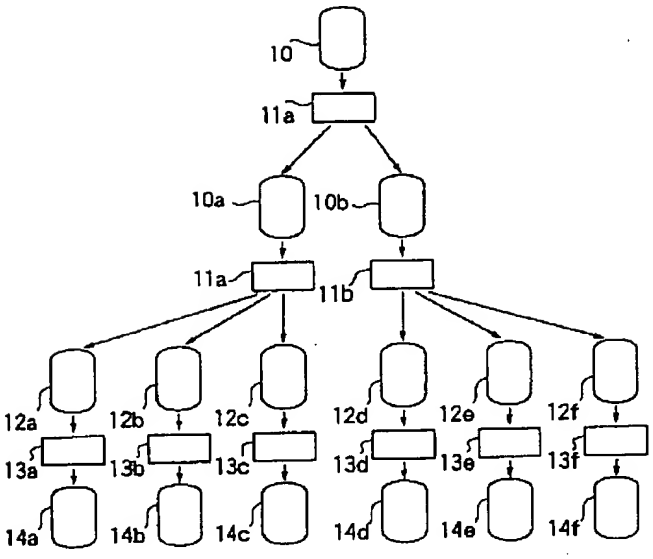
【図５】


開　始
ソースプログラム解析
S01
S02
関数分析
S03
S04
S05
LISP に変換
FORTRAN に変換
C に変換
S06
N
全モジュール終了
Y
終　了

【図６】


10
データＡを他計算機より受ける
10a
データＡについて記号処理
10b
データＡについて数値演算を行う
10c
処理結果を表示、記憶する
10d

【図８】


10
11a
10a
10b
11a
11b
12a
12b
12c
12d
12e
12f
13a
13b
13c
13d
13e
13f
14a
14b
14c
14d
14e
14f